



Software Improvement Group



Certification of Technical Quality of Software Products

Joost Visser

30 March 2010

Arent Janszoon Ernststraat 595-H
NL-1082 LD Amsterdam
info@sig.nl
www.sig.nl

Software Quality Waves



1970s

- Higher-order languages

1980s

- Design methods and tools (CASE)

1990s

- Process

2000s

- Product

2010s

- ?

2 | 23



“Focus on the product to improve quality”

“The CMM [..] is not a quality standard”

Bill Curtis

co-author of the *Capability Maturity Model (CMM)*,
the *People CMM*, and the *Business Process Maturity Model*.
In “CMMI: Good process doesn't always lead to good quality”
Interview for TechTarget, June 2008
<http://tinyurl.com/process-vs-quality>

Software Improvement Group

Background, activities

Software Product Quality

ISO/IEC 9126

SIG quality model

Applications

Assessment Improvement Certification



Background

4 | 23

- Spin-off from CWI in 2000, self-owned, independent
- Strong academic background, innovative, award-winning, profitable

Activity

- Management advisory, fact-based
- Accredited software analysis lab employs analysis tools and models
- Experienced staff transforms analysis data into advice

Track record

- Finance, government, logistics, telecom, manufacturing, energy, ...
- We analyze over 90 systems annually

Who is using our services?



Financial



Public



Logistics



IT



Other





Software Risk Assessment

- In-depth investigation of software quality and risks
- Answers specific research questions



Software Monitoring

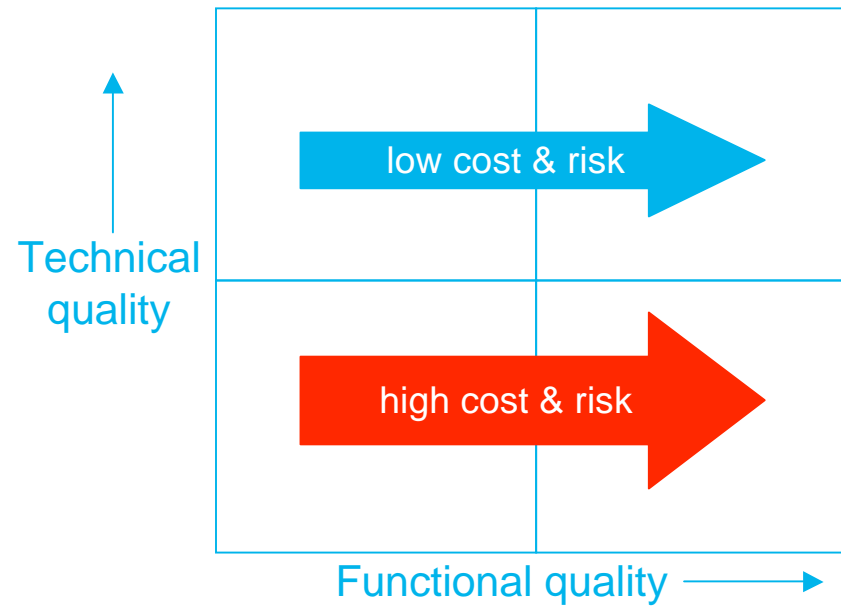
- Continuous measurement, feedback, and decision support
- Guard quality from start to finish



Software Product Certification

- Five levels of technical quality
- Evaluation by SIG, certification by TÜV Informationstechnik

Functional vs technical quality

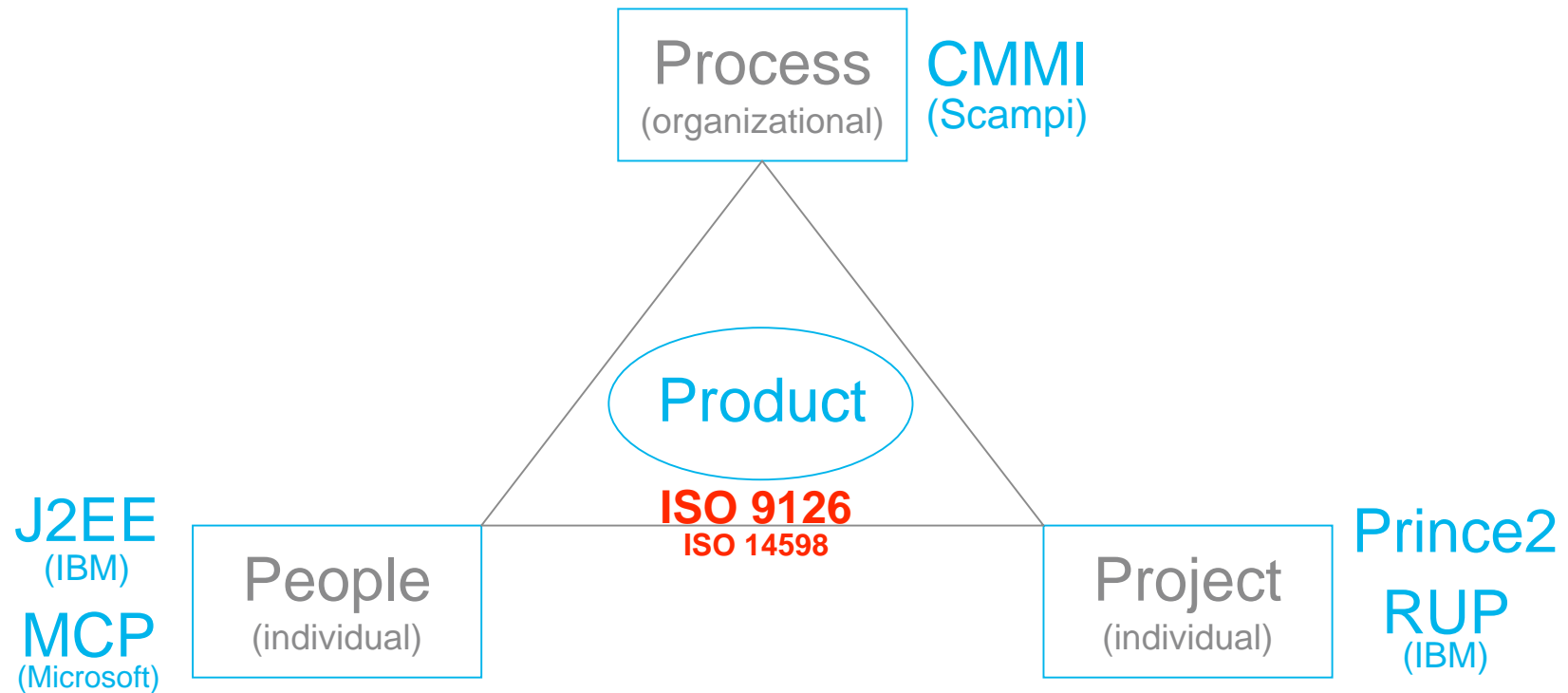


Software with high technical quality can evolve with low cost and risk to keep meeting functional and non-functional requirements.

The Bermuda Triangle of Software Quality

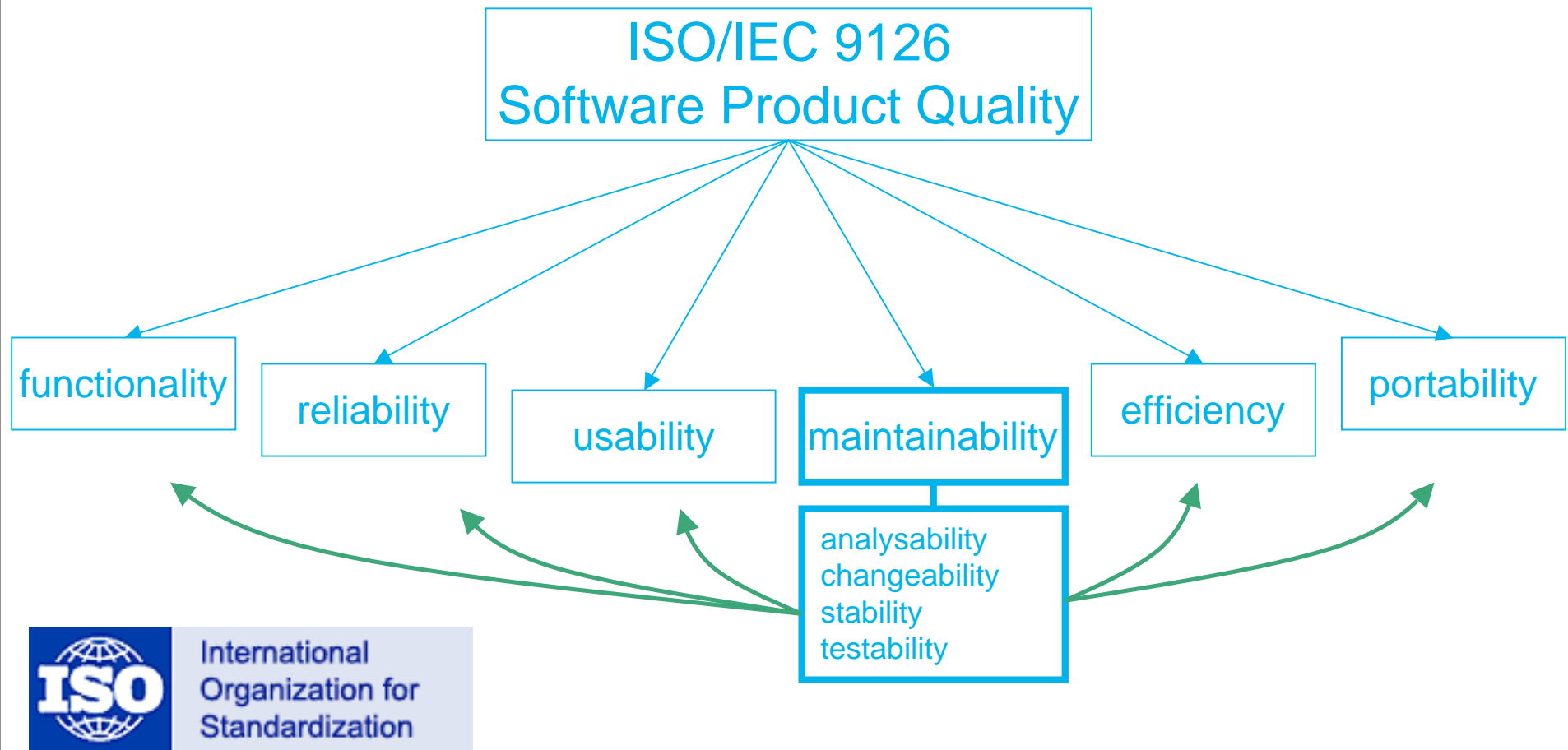


8 | 23



ISO/IEC 9126, Part 1

Software product quality characteristics

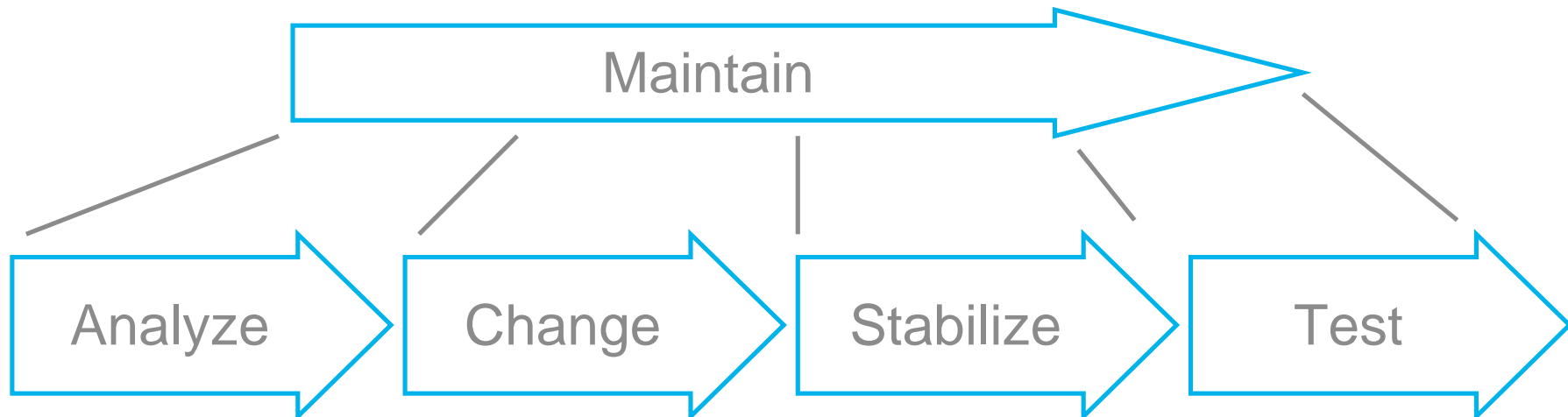


ISO/IEC 9126: Software Engineering - Product Quality

10 | 23

Maintainability =

- *Analyzability*: easy to understand where and how to modify?
- *Changeability*: easy to perform modification?
- *Stability*: easy to keep coherent when modifying?
- *Testability*: easy to test after modification?



SIG Quality Model

How do measurements lead to ratings?



A practical model for measuring maintainability

11 | 23

Heitlager, Kuipers, Visser in QUATIC 2007, IEEE Press

- a. Aggregate measurements into “Quality Profiles”
- b. Map measurements and quality profiles to ratings for system properties
- c. Map ratings for system properties to ratings for ISO/IEC 9126 quality characteristics
- d. Map to overall rating of technical quality



Quality roadmap

12 | 23

- “final product shall be 4 stars” in development project
- “complexity from 2 to 4 stars by 3rd month” in maintenance project

Dashboard

- Regular analysis of source code typically once per week
- Shown on dashboard with overviews and drill down possibilities

Consultancy

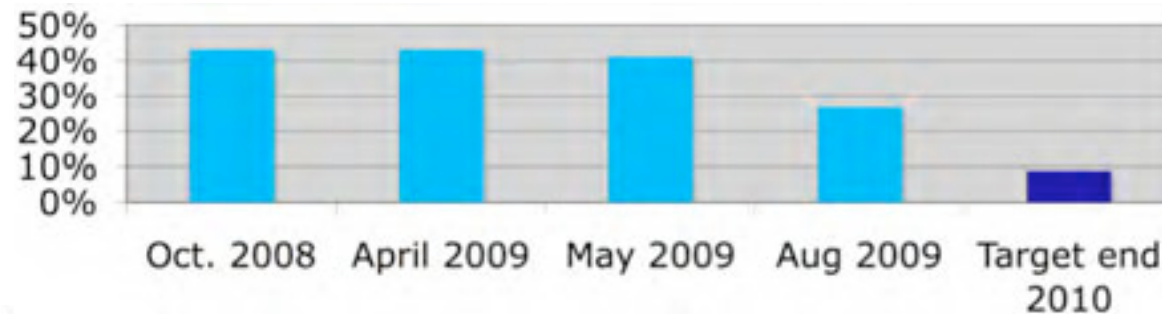
- Regular reports in control? on schedule? within budget?
- Guard quality agreements, meet quality targets from “trust me” to “show me”

Software Monitor

Example: vendor management and roadmap

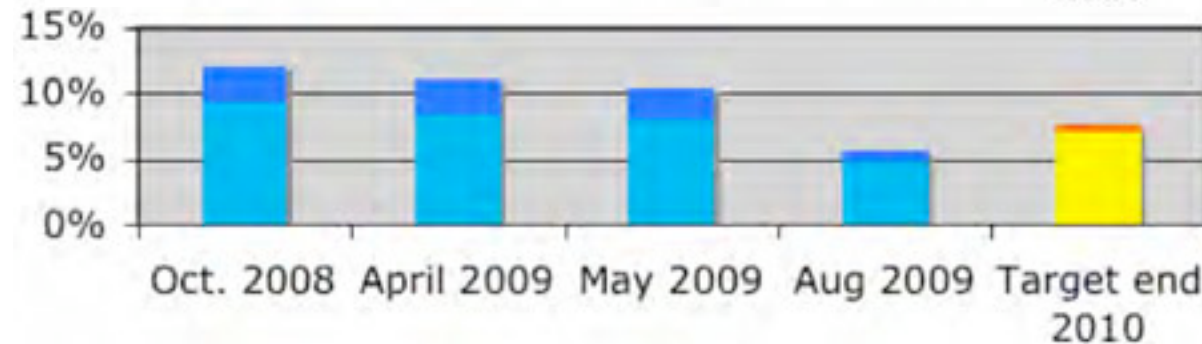


Duplication



13 | 23

Complexity



From client testimonial:

- “Technical quality: as it improves adding functionality is made easier”
- “As quality was increasing, productivity was going up”

Certification

14 | 23

- Third party gives written assurance that a product conforms to specific requirements

Essential elements

- Criteria: based on ISO/IEC 9126
- Evaluation body: institute that examines the product
- Certification body: institute that confirms evaluation process and result

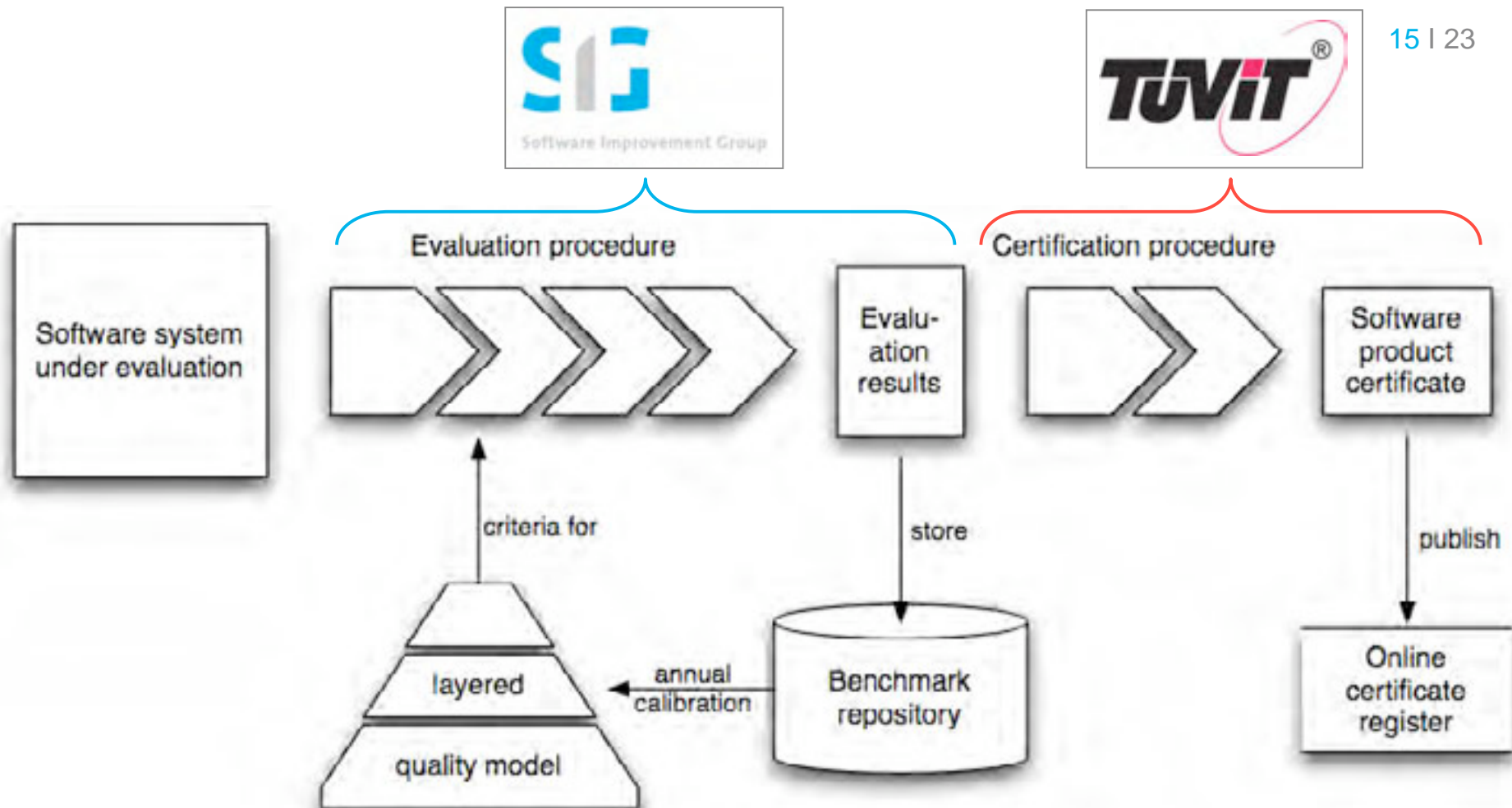


Results

- Evaluation report, including measurements
- Certificate and quality mark:
“TÜViT Trusted Product *Maintainability*”



Overview



Evaluation Criteria

Calibrated against benchmark repository



Eligibility for quality mark

- High-level description: fulfill minimal requirements
- Quality ratings: 2 stars or more
- Overall rating: 3 stars or more

16 | 23

Calibration w.r.t. SIG Benchmark Repository

- At level of property ratings
- Against large set of systems
- Multiple technologies, multiple domains
- E.g. about 5% of all systems reach 5 stars for the *complexity* property

| | |
|-------|-----|
| ☆☆☆☆☆ | 5% |
| ☆☆☆☆☆ | 30% |
| ☆☆☆☆☆ | 30% |
| ☆☆☆☆☆ | 30% |
| ☆☆☆☆☆ | 5% |

| High-level description |
|------------------------|
| ✓ ✓ ✓ |
| Quality Ratings |
| ☆☆☆☆☆ |
| ☆☆☆☆☆ |
| ☆☆☆☆☆ |
| ☆☆☆☆☆ |
| Overall Rating |
| ☆☆☆☆☆ |

Software Product Certification

Who uses and how?



Issued certificates

17 | 23

| | | |
|----------|---------------------------------|--------------------------|
| Kas BANK | Tri-party collateral management | internal development |
| Rabobank | Bank-lobby console CRM | Ordina / Cognizant India |
| ProRail | On-board track visualization | Sogeti |
| KLM | Transfer kiosk | Accenture |

and 4 to be issued during ceremonial event on April 22
and 3 evaluations underway

Current applications of SIG/TÜViT evaluation criteria

- Meet criteria before **acceptance** or **deployment**
- Define improvement **roadmaps** towards certifiability
- Include criteria in **RFPs**, **contracts**, and **SLAs**

Risk Mitigation in the Software Lifecycle

Which instrument should be used when?



Objectives

18 | 23

- Detect and resolve risks as early as possible. Measurable progress and quality.
- Clear assignment of responsibility.



- Smooth project execution.
- Successful outcome, both for client and supplier.

Risk Mitigation in the Software Lifecycle

Which instrument should be used when?

Pre-contractual phase

19 | 23

- Formulate high-level requirements. Set technical constraints.
- Tender situation: request, then select proposal.



**Contract
+ Plan**
balanced
feasible
enforceable

- Include audit right.
- Set quality targets.
- Agree on bonus / penalty clause.

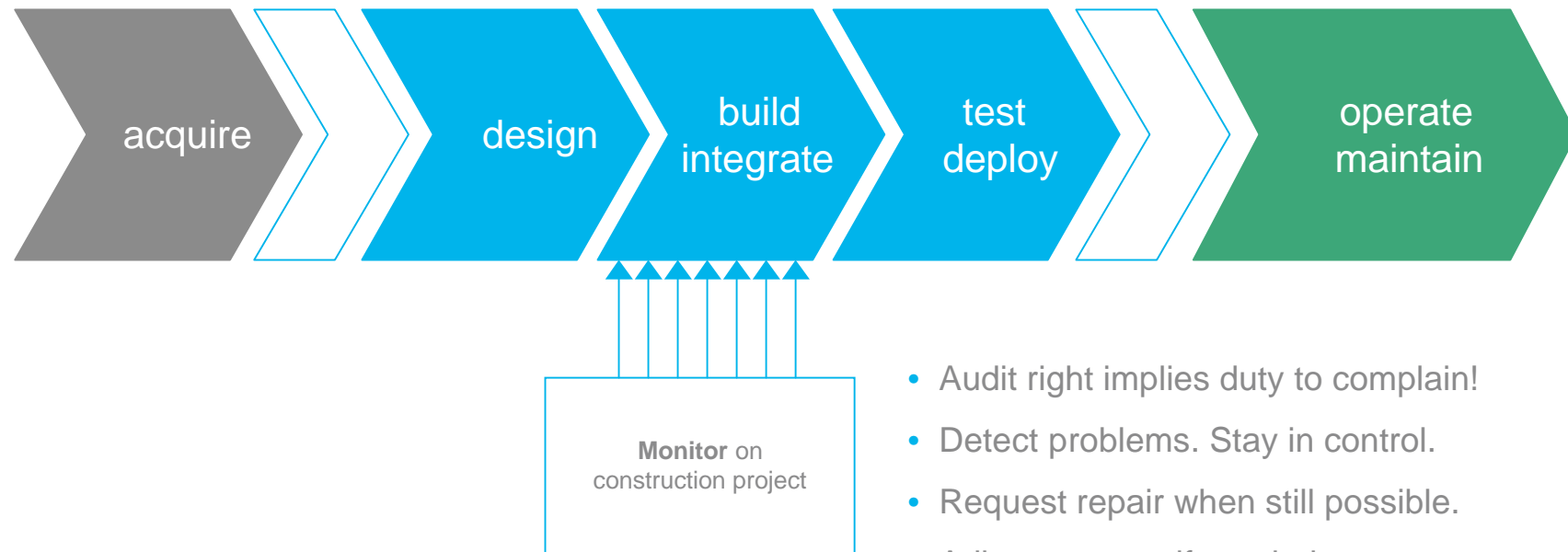
Risk Mitigation in the Software Lifecycle

Which instrument should be used when?

Construction phase

20 | 23

- Produce and/or adapt software.
- Structured process. For example by milestones or releases.



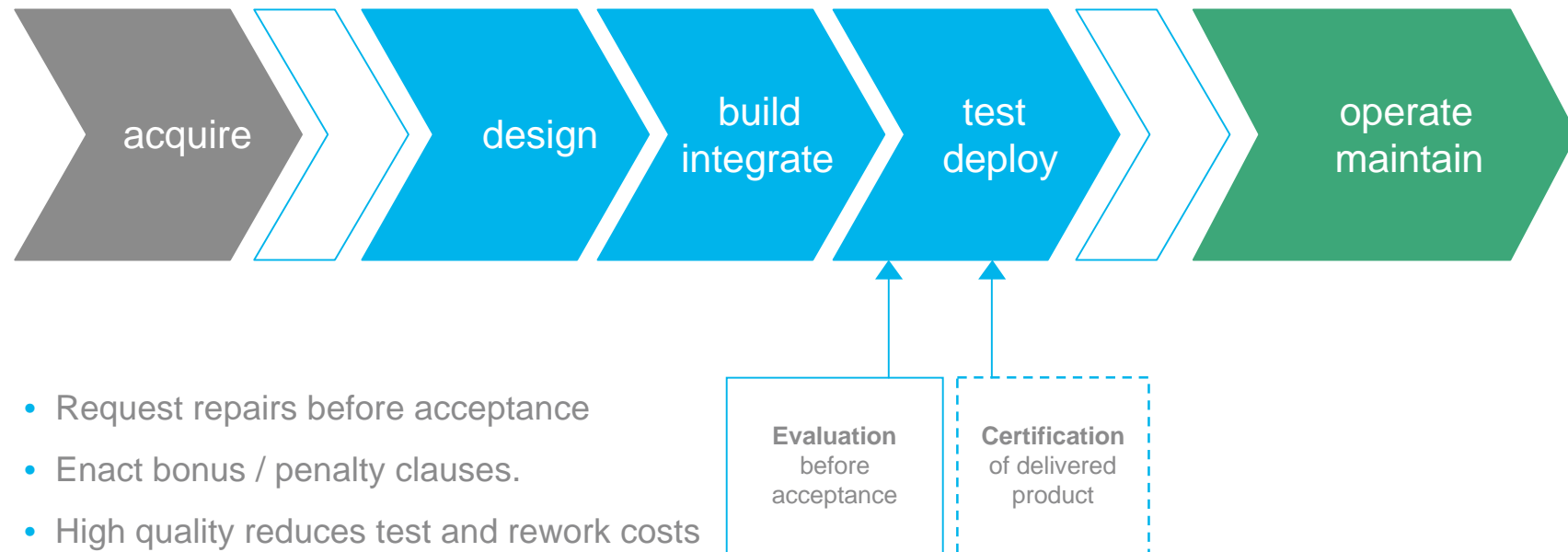
Risk Mitigation in the Software Lifecycle

Which instrument should be used when?

Test and deployment phase

21 | 23

- Acceptance testing.
- Deployment to production environment.



Risk Mitigation in the Software Lifecycle

Which instrument should be used when?



22 | 23

Test and deployment phase

- Acceptance testing.
- Deployment to production environment.



Certification:

- Written assurance of quality.
- Establish future-proofness.
- Advertise success.

Product
delivered
certified
maintainable

Conclusion



Technical quality of software can be defined and measured

23 | 23

- ISO/IEC 9126 provides definitions
- SIG quality model performs quantification and rating

Measurement used to ...

- Set technical requirements
- Monitor quality and progress
- Certify products

To help achieve ...

- Vendor management
- Project success
- Portfolio rationalization

Thank you!



Software Improvement Group

24 | 23



Dr. ir. Joost Visser

j.visser@sig.eu

<http://twitter.com/jstvssr>

www.sig.eu

+31 20 314 0950